PCT/IB05/555

Europäisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

04290597.6

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Europäisches Patentamt
European Patent Office
Office européen des brevets

Anmeldung Nr:
Application no.: 04290597.6
Demande no:

Anmeldetag:
Date of filing: 04.03.04
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Schlumberger Systèmes
50, avenue Jean Jaurès
92120 Montrouge
FRANCE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

A secure sharing of resources between applications in independent execution environments in a portable device

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06F9/46

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PL PT RO SE SI SK TR LI

En date du 29 juillet 2004 (29.07.2004), le nom de la demanderesse a été changé comme suit:

Axalto S.A.
50, Avenue Jean Jaurès
92129 Montrouge
(France)

A secure sharing of resources between applications in independent execution environments in a portable device (e.g. smart card)

Integrated Circuit Cards (IC cards or 'smart cards') are intrinsically secure computing platforms ideally suited to providing enhanced security and privacy functionality to applications. They are also being used in the Wireless phones, and other communication devices, as a place to store user subscription data, user private keys and other private or confidential data. They provide a mean for secure storage and computational facilities for sensitive information such as:

- Private keys and key fragments.
- Account numbers and stored value.
- Passwords and shared secrets.
- Authorizations and permissions.

At the same time, many of these tokens provide an isolated processing facility capable of using this information without exposing it within the host environment where it is at potential risk from hostile code (viruses, Trojan horses, and so on). This becomes critically important for certain operations such as:

- Generation of digital signatures, using private keys, for personal identification.
- Network authentication based on shared secrets.
- Maintenance of electronic representations of value.
- Portable permissions for use in off-line situations.

Current smart cards use the communication protocol defined in the ISO 7816 standards by which an asynchronous protocol is being used and APDU commands carry application level information. This protocol is also being used in mobile phones, and the GSM and 3GPP standards conform to it.

Today there are additional and more rapid synchronous communication protocols that are being integrated in new smart cards. This allows the addition of a synchronous communication protocol (e.g. USB or other), in parallel to the ISO 7816 communication protocol. Since each communication channel uses a different set of pin contacts there are no inter-dependencies between the two.

Today the terminal (phone or other communication device) uses the smart card services by sending specific commands (called APDU commands) that invoke different computation services or cause the retrieval of data. The smart card application performs the request and returns the needed data.

New cards integrate an additional and independent communication protocol (e.g. USB) on a different set of pin contacts of the card. In this case the card manages the two communication channels by two independent processes.

The above card interface can be managed by defining two virtual execution environments:

- APDU execution environment
- New execution environment

The "APDU execution environment" is the existing (legacy) execution environment to which all existing standards apply. For example, for the SIM card, it concerns all the standards that define the applications and services that the card implements for network authentication, SIM-Toolkit applications etc.

The "New execution environment" is independent of the old one ("APDU execution environment") and does not have any communication with it. This is important in order to assure that current card applications are not being affected. This backward compatibility need imposes a separation (fire-walling) between applications in the two execution environments. These execution environments can be illustrated in figure 1.

The "New execution environment" can allow the execution of several applications, as is the case for the "APDU execution environment". An example is a Web server that may integrate several web applications.

The "New execution environment" can implement a new set of applications that are independent of the legacy applications in the "APDU execution environment". However, the card issuer may want to allow a certain level of sharing of data and operating system services that will not interfere with the behavior of the legacy environment ("APDU execution environment"). A communication protocol between applications in the two execution environments may also be implemented in order to allow a secure sharing of data or functions.

This principle can be extended to more than two protocol stacks. Even if the portable device has only two physical layers, these last ones can be shared between several protocol stacks.

In a particular case, a physical layer can provide a support for several logical channels (or "pipes"). So, to illustrate this concept, the USB protocol is able to support several logical pipes

(end points) on the same physical medium. Some logical pipes can be dedicated to a protocol stack and some others to another one. The concept is named " composite device" within the USB environment.
Through the same USB connector, the host can manage several types of device (e.g. mouse, keyboard).

The concept applied to a smart card able to host an USB interface can take advantage of the invention. Through this USB layer, we can imagine host at the card level:

- A TCP/IP stack
- A mass storage stack

Beside this USB communication, the smart card uses the regular ISO7816 protocol to establish another link of communication. This example shows that at least three different protocol stacks can run in the same smart card. So, the invention is applicable where in each specific context, an application can establish a secure and controlled bridge between execution environments running different protocol stacks.

More generally, the portable device can be a smart card or any secure portable device able to host at least one physical channel of communication and wherein at least one logical channel of communication can be opened. The unique condition is to host at least two logical channels of communication whatever is the number of physical channel of communication. Consequently, the invention can be usefully used within smart card running an ISO 7816 protocol but with multiple logical channels or with a Multi Media Card or a dongle (e.g. USB dongle).

The invention is also applied, but not limited to, the following scenarios, which illustrate well the ability to have a combination of independent physical and logical communication channels:

- A portable device that has one USB communication channels but with several logical channels ("pipes") where APDU commands are sent on one logical channel, to address the legacy applications, and a TCP/IP protocol stack runs on the other logical channel.
- A portable device that has one or more physical or logical channels, each associated with a different isolated execution environment. The physical communication channels can be, but are not limited to, the following examples:

- Multi Media Memory card (MMC) protocol
- SPI (Serial Peripheral Interface) protocol
- USB protocol
- Smart card contactless protocols
- ISO 7816 protocol
- ISO (FCD)15693 protocol
- ISO 14443 protocol
- The communication protocol defined in the TS 102.221 standard

The sharing of data and operating system services between the multiple execution environments may rely on several mechanisms and we consider only one couple in a set of couples:

- A pipe between two applications in the two execution environments when one is the data producer and the other is the data consumer
- A file sharing when one application has read access to a file and the other has a read/write access to the same file (sort of implementation of a pipe when one application is the producer and the other is the consumer)
- A communication protocol that is defined and implemented internally by the card Operating System and is shared by the two applications
- Sharing of re-entrant functions and function libraries of the card operating system

See figure 2 illustrates the interactions between the two execution environments:

Figure 2, App2 in the "New execution environment" can share information and/or services with App A & B in the "APDU execution environment". The applications are not necessarily active at the same time. It may be that App A and/or App B were invoked and produced some data that may then be used by App 2 when it starts to run.

The smart card underlining operating system offers the resources and data sharing mechanisms of the following types:

- File sharing controlled by Access Control List (ACL)
- Stream based communication (data pipe) controlled by Access Control List (ACL)
- Proprietary communication mechanisms between applications which satisfy the following characteristics:

    - Enables to send and receive data between two applications running in two different execution environments
    - The access to this communication mechanism is controlled by Access Control List (ACL)
- Re-entrant functions that are published by shared libraries in the card underlining operating system
- Re-entrant functions that are published by an application running in one execution environment to an application running in the other execution environment (e.g. RPC like)

Access Control List (ACL)

Access Control List (ACL) is a mean to identify an application, or the entity that invoked the application, and attach access rights to it. An ACL can be represented as a pair of the following items:

<id, access conditions>

The id can be one of the following:

- Application id in the execution environment
- User id for whom the application is performing a task
- External entity for whom the application is performing a task (e.g. card administrator or super user)

The access conditions may be, but not limited to one of the following:

- Read
- Write
- Execute
- Any combination of the above

The card operating system may offer shared resources to the two execution environments. An application will have access rights to use the shared resources, if there is an ACL that defines its access rights to it. The shared resources may be a communication mechanism between the two execution environments or be a set of shared functions. Each application may be granted the rights to use the shared resources if it satisfies the corresponding access conditions (ACL) attached to each resource.

Card administrator role

The ACLs are defined by an entity that is called card administrator. Normally, this is the card issuer or "super user". This entity can define and change ACLs in the card for the sharing of resources between the execution environments.

The identity of the "super user" is normally proved by cryptographic means that provide a proof of possession of an administrator key.

Example 1

Lets take the example of a SIM card which also has the additional "New execution environment". The SIM card implements all the services that are defined in the related GSM standards and they all run in the APDU execution environment. The APDU execution environment communicate with the mobile phone via the ISO 7816 and GSM standardized protocols.

The "New execution environment" communicates with the mobile phone via a USB protocol with TCP/IP and HTTP on top, and runs an HTTP web server with an application that can perform the following task:

- Receive information about a content that is installed in the mobile phone
- Compute the permissions to execute a content that is installed in the mobile phone (a Digital Rights management application)

For this purpose the application needs to get a license information from a file. This license was updated in the card via an OTA message (Over the Air protocol), which is a protocol that is defined in the GSM standards. An application that was running in the APDU execution environment received this message and updated the file accordingly. The GSM application can update the file since there is an ACL that gives it a read-write permission to this file.

In order to get the license information the Web application in the "New execution environment" reads the shared file in which the license is stored. The application can read the shared file since there is an ACL for it that gives it a read-only permission to this file. If it will try to also write to this file the operating system will not allow it and will throw an exception.

The Web application that runs in the "New execution environment" also needs to perform a decryption of the content that needs to be rendered in the mobile phone. For that it needs to access a library that performs this decryption and that uses a key that was personalized in the card during manufacturing or updated OTA (Over the Air protocol). This application can execute the decrypt function in the decryption library since there is an ACL that gives it an "execution" permission" to this shared function.

See figure 3.

The above figure illustrates the communication mechanisms between the two applications. The dotted lines indicate that the applications can exchange data between them or share some common functions. The actual communication of the data is done by file sharing or by calling shared libraries that are implemented by the operating system.

Claims:

1. A portable device comprising: at least one physical channel of communication to at least one apparatus wherein each physical channel of communication can host at least one logical channel of communication (pipe) and for each logical channel is associated a secure/independent execution environment when more then one logical channel is being used.

2. A portable device as recited in the claim 1 wherein the portable is a smart card.

3. A portable device as recited in the claim 1 wherein the portable is a Multi Media Memory card.

4. A portable device as recited in the claim 1 wherein the apparatus is a mobile handset.

5. A portable device as recited in the claim 1 wherein the apparatus is a personnel computer.

6. A portable device as recited in the claim 1 wherein at least one of the physical channel of communication uses the USB protocol.

7. A portable device as recited in the claim 1 wherein at least one of the physical channel of communication uses the SPI protocol

8. A portable device as recited in the claim 1 wherein at least one of the physical channel of communication uses the MMC protocol

9. A portable device as recited in the claim 1 wherein at least one of the physical channel of communication uses a protocol for contactless smart card

10. A portable device as recited in the claim 9 wherein the protocol of communication is defined in the ISO (FCD)15693

11. A portable device as recited in the claim 9 wherein the protocol of communication is defined in the ISO 14443.

12. A portable device as recited in the claim 1 wherein at least one of the physical channels of communication uses the protocols defined in the TS 102.221 standard.

13. A portable device as recited in the claim 1 wherein at least one of the physical channels of communication uses the protocols defined in the ISO7816 standard.

14. A portable device as recited in the claim 1 wherein each physical channel is independent of the others.

15. A portable device as recited in the claim 1 and at least one of the claims from 2 to 14 wherein for execution environments for each of the logical communication channel; with applications that can be executed independently in each execution environment, in an isolated manner from the other execution environment but with some resources that can be shared between applications, in the different execution environments, based on access condition lists (ACLs).

16. A portable device as recited in claim 1 and 15 for which the resource that can be shared between applications in the different execution environments is a shared file for which access conditions (ACLs) are defined for the applications that can access it to specify the rights of the applications to perform operations on this file (e.g. read, write etc.).

17. A portable device as recited in the claim 1 and 15 for which the resource that can be shared between applications in the different execution environments is a shared object called "pipe" on which it is possible to write data in a "first in first out" (FIFO) manner and for which access conditions (ACLs) are defined for the applications that can access it to specify the rights of the applications to perform operations on this pipe (e.g. put, get etc.).

18. A portable device as recited in the claim 1 and 15 for which the resource that can be shared between applications in the different execution environments is a shared function that is implemented by the underlining common operating system and for which access conditions (ACLs) are defined for the applications that can access it to specify the rights of the application to invoke it.

19. A portable device as recited in the claim 1 and 15 in which an applications in an execution environments can share some functions with an application in another execution environment by allowing the other application to invoke these functions and where access conditions (ACLs) are defined for the application that can access this shared functions.

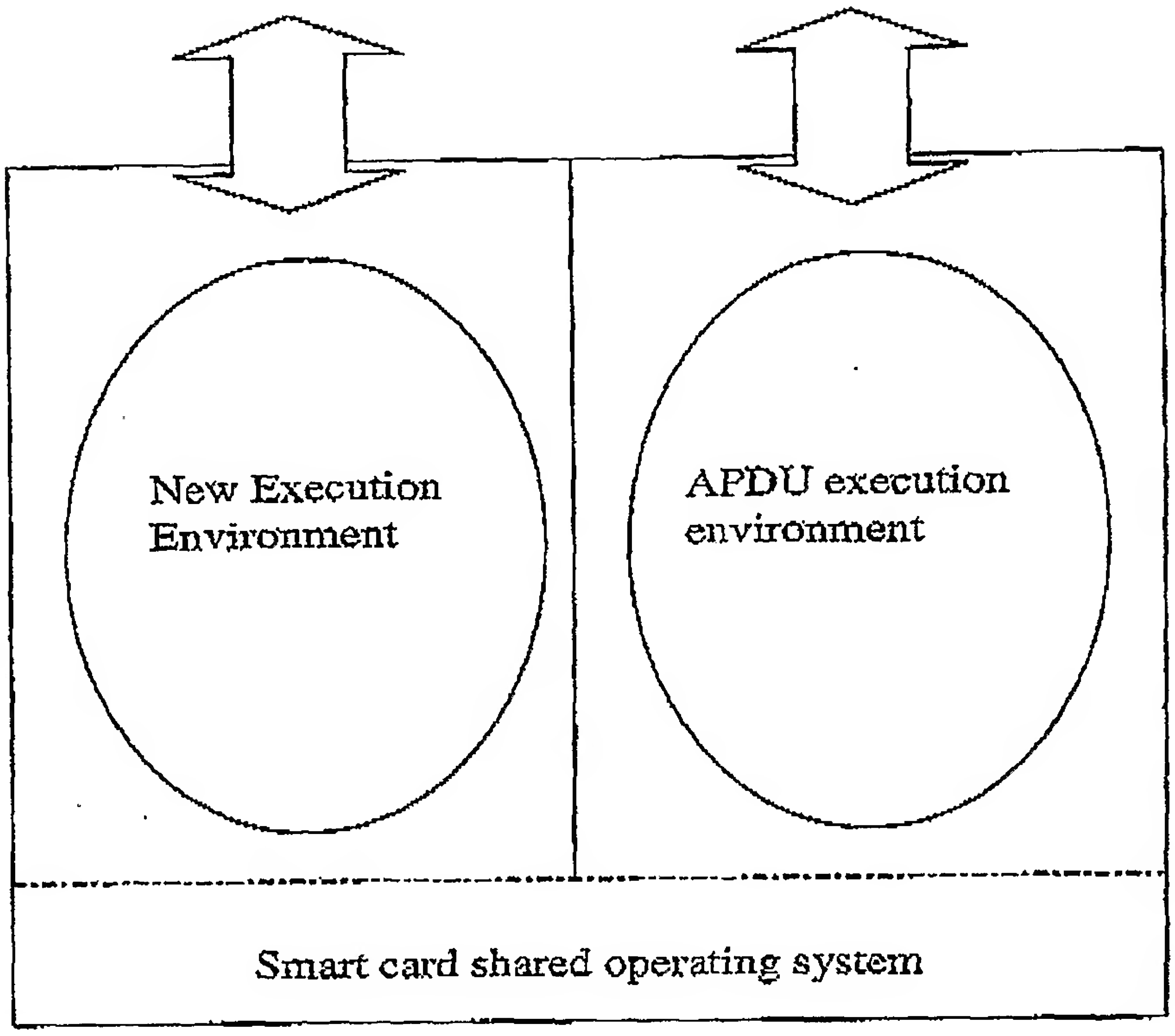
New Execution Environment
APDU execution environment
Smart card shared operating system

Figure 1

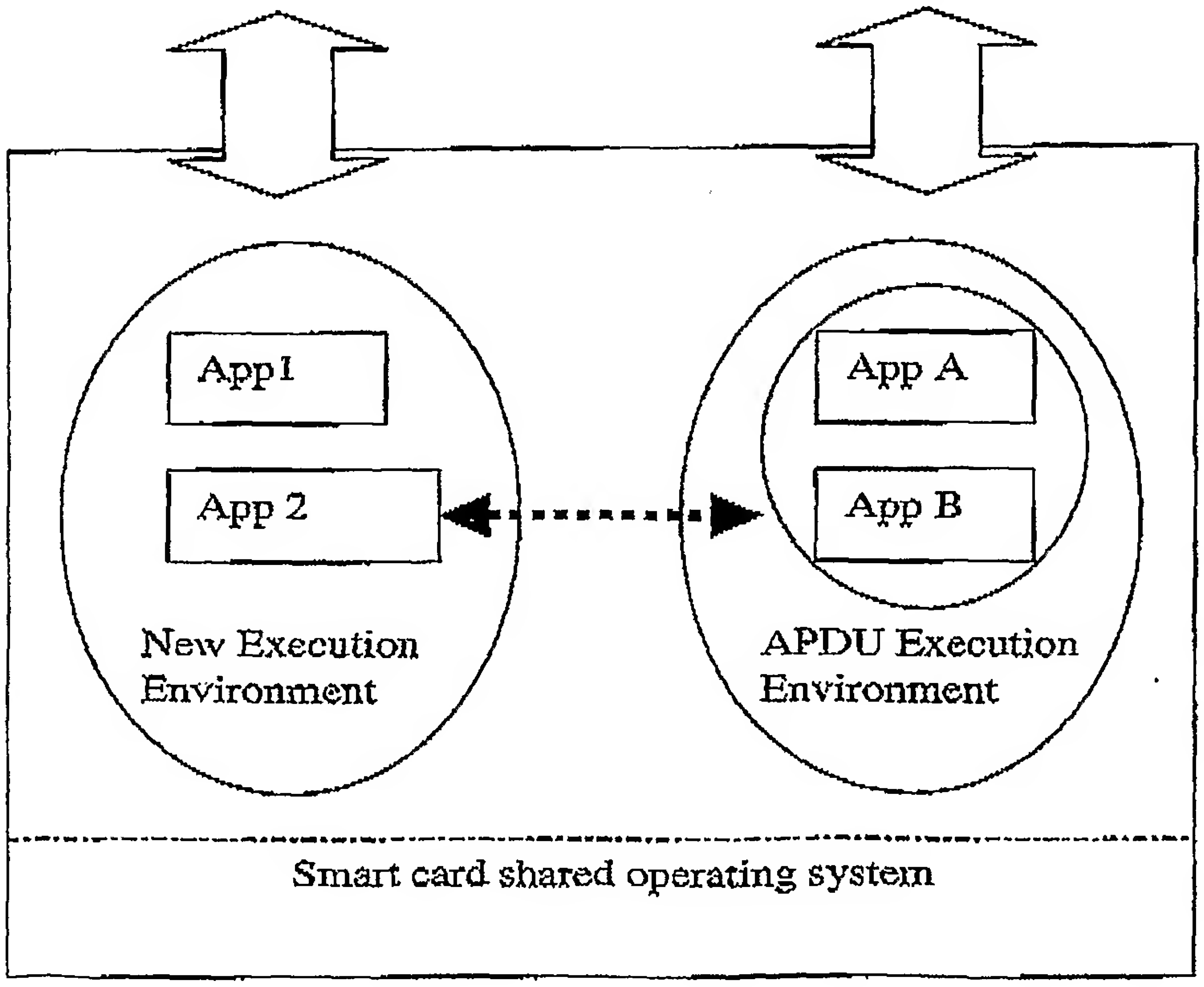
App1
App 2
App A
App B
New Execution Environment
APDU Execution Environment
Smart card shared operating system

Figure 2

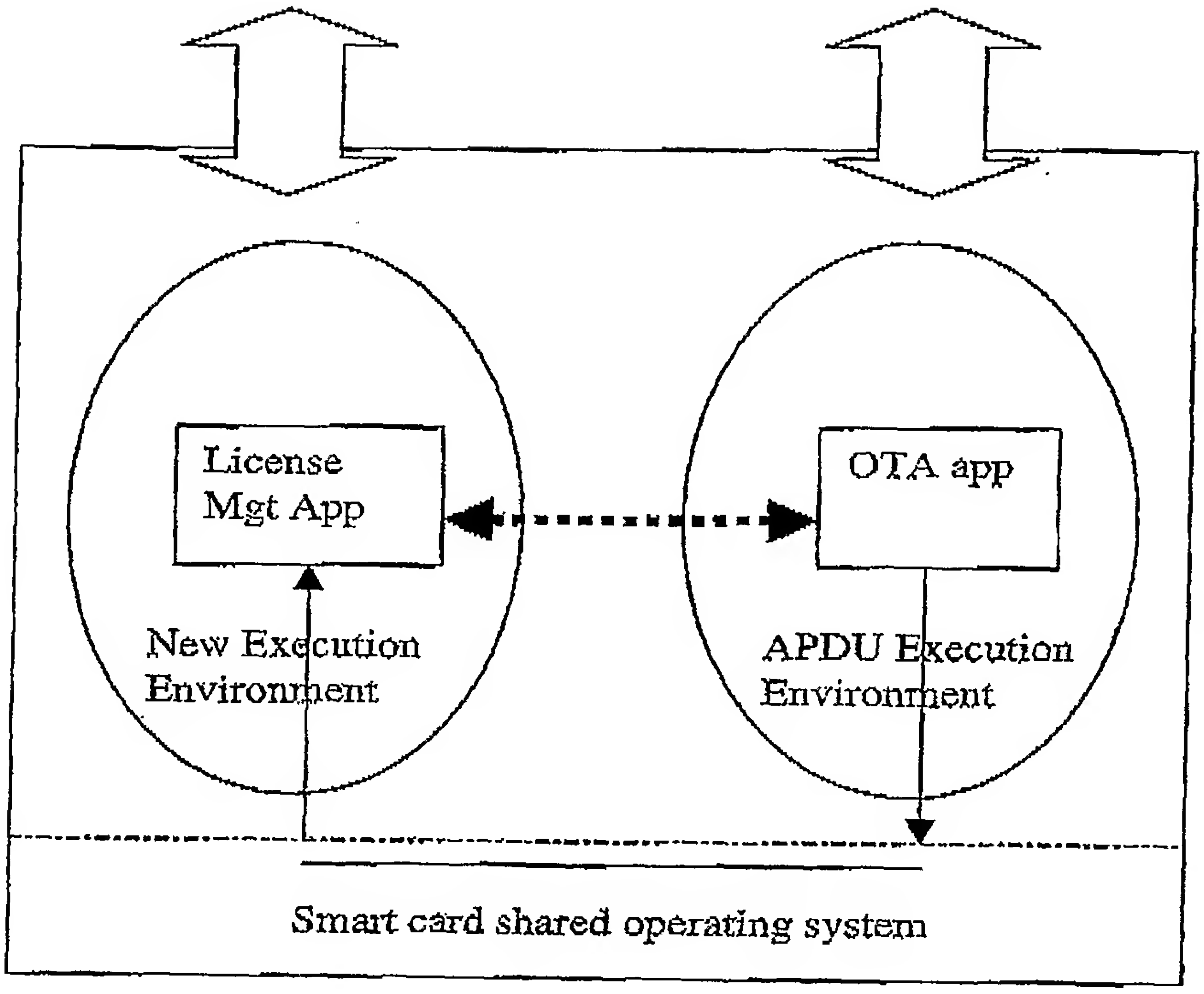
License
Mgt App
OTA app
New Execution
Environment
APDU Execution
Environment
Smart card shared operating system

Figure 3